Máster Universitario en Lógica, Computación e Inteligencia Artificial.

Aprendizaje Automático

Diego Enrique Morales Polanco

Cuestionario 3

**Ejercicio 1.** Sea un perceptrón con dos neuronas en la capa de salida que llamaremos A y B, sea *softmax* la función de activación de esa capa de salida. A partir de vamos a construir un nuevo perceptrón multicapa haciendo los siguientes cambios.

1. En la función de activación de A y B será la función identidad. .
2. Tras la capa formada por A y B, en añadimos una nueva capa formada por dos neuronas, que llamaremos C y D. La función de activación de C y D es la función sigmoide.
3. Los pesos asociados a las entradas ficticias de C y D, a los pesos de las conexiones entre las neuronas indicadas. Supondremos también que y .

Supongamos que, para cualquier entrada, las redes y proporcionan la misma salida (esto es, la salida de la neurona A en es la misma salida que la neurona C en después de aplicar las respectivas funciones de activación y, análogamente, las salidas de B en y D en después de aplicar las funciones de activación también coinciden). Calcular y .

Primero dibujamos ambos perceptrones descritos para tener una visualización de las redes:

A diagram of a softmax

Description automatically generated A diagram of a network

Description automatically generated

En N2, la salida de A y B es y respectivamente debido a la función identidad.

Según el enunciado, N1 y N2 proporcionan las mismas salidas. Por lo que:

Aplicando la función softmax para N1, tendríamos las siguientes expresiones de las salidas:

Expandiendo N2 obtenemos lo siguiente:

Expandiendo la función sigmoide:

Sabiendo que y los bias son 0.

Recordando que:

Tenemos las siguientes ecuaciones:

Resolviendo:

Aquí vemos que para que el lado izquierdo sea igual que el lado derecho, debe ser igual a :

Exactamente el mismo procedimiento se puede hacer para la otra igualdad, finalmente:

**Ejercicio 2.** Supongamos que tenemos un conjunto de datos donde no hay ruido (noise) y un algoritmo de aprendizaje L tal que L(D) siempre es la misma hipótesis, sea cual sea el vector de entrenamiento D. ¿Qué podemos decir sobre el error esperado, el sesgo y la varianza del algoritmo L sobre un conjunto de test extraído de ese conjunto de datos?

En primera instancia, ya que no hay ruido, podemos predecir que el error esperado del algoritmo dependerá únicamente de el sesgo y la varianza. Ahora procederemos a analizar la influencia del sesgo y la varianza.

Si el resultado del algoritmo de aprendizaje siempre es la misma hipótesis independientemente del vector de entrenamiento, significa que el predictor esperado , es igual a la hipótesis que resulta del algoritmo de aprendizaje en cualquier vector de entrenamiento, . Esto significa que:

El valor que el predictor le asigna a cada punto x de cualquier conjunto de entrenamiento es valor que le asignaría el predictor esperado.

Tomando esto en cuenta, podemos afirmar lo siguiente a partir de la definición de varianza:

La varianza en cualquier vector de prueba extraído es 0.

Si se toma un conjunto de entrenamiento con todos los puntos , el predictor resultante será el mismo que tomando cualquier subconjunto de con menor tamaño que Esto significa que, si el algoritmo de aprendizaje es consistente con el conjunto de entrenamiento que tiene todos los elementos de X, pues clasificará correctamente todos los elementos , independientemente de el conjunto de entrenamiento que se le proporcione para ser entrenado y el conjunto de prueba que se extraiga. Por lo que el sesgo sería 0.

Si el algoritmo de aprendizaje no es consistente, todo el error esperado dependerá únicamente del sesgo introducido por el algoritmo.

Se puede afirmar que el algoritmo tiende a hacer underfitting al vector de entrenamiento D.

**Ejercicio 3.** Supongamos que la asignación de etiquetas a los elementos de cualquier conjunto de entrenamiento es determinista, esto es, al mismo siempre le asignamos el mismo valor de clasificación . Consideremos un algoritmo de aprendizaje tal que a cada conjunto de entrenamiento le asigna , esto es, con la propiedad de que el valor esperado de al variar es la misma etiqueta que tenemos para en el conjunto de entrenamiento. ¿Qué podemos decir sobre el sesgo y la varianza de ese algoritmo de aprendizaje sobre un conjunto de test extraído de ese conjunto de datos?

Primero, el hecho de que al mismo siempre se le asigne el mismo valor de clasificación , significa que no hay ruido introducido por el conjunto de entrenamiento, ya que la etiqueta esperada de cada dato () es la misma que la etiqueta real del dato .

Por otro lado, si para cada que se obtenga del algoritmo de aprendizaje al variar , el valor de clasificación asignado por a cada elemento del conjunto de entrenamiento es igual al valor real de clasificación, significa que es un algoritmo consistente con su conjunto de entrenamiento y lo suficientemente expresivo para, valga la redundancia, expresar todos los conceptos en .

En base a esto, del sesgo se puede concluir lo siguiente:

* Todo del conjunto de entrenamiento que también se encuentre en el conjunto de test será clasificado correctamente por .
* Que el algoritmo sea consistente con el conjunto de entrenamiento no significa que lo será con el conjunto de test, por lo que habrá sesgo introducido por el algoritmo de aprendizaje.
* Si todos los elementos del conjunto de test aparecen en el conjunto de entrenamiento el sesgo será 0. Por lo que en este caso la distribución entre el conjunto de entrenamiento y test tiene un efecto en el sesgo.

De la varianza se puede concluir lo siguiente:

* El predictor esperado ( depende del conjunto de conjuntos de entrenamiento elegido. Si un elemento del conjunto de prueba se encuentra en el conjunto , será clasificado correctamente ya que será el valor de la etiqueta de .
* Si se encuentra en el conjunto de conjunto de entrenamientos posibles pero no en el conjunto de entrenamiento D elegido, pues habrá una varianza debido a la elección de un conjunto de entrenamiento u otro ya que será distinta a .
* Se puede afirmar que el algoritmo tiende a hacer sobreajuste al vector de entrenamiento D.

**Ejercicio 4.** Si tomamos la función identidad ) como función de activación, entonces el perceptrón simple (esto es, una única neurona) es capaz de aprender la función booleana binaria Si la respuesta es **Verdadero** tienes que dar un perceptrón que caracterice a . Si la respuesta es **Falso**, tienes que dar las razones de por qué no puede existir tal perceptrón.

La respuesta es **Falso.**

A black circle with black lines and letters

Description automatically generated

De la forma general de las neuronas:

De la tabla de verdad de la compuerta AND:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Nos quedan las siguientes ecuaciones:

Se puede apreciar que para las primeras 3 ecuaciones, los valores que la satisfacen ( no pueden satisfacer la ecuación 4. Por esto, no se puede usar un perceptrón simple con la función identidad como función de activación para simular el comportamiento de una puerta lógica AND.

**Ejercicio 5.** Consideremos la red neuronal de la siguiente figura donde tomamos la función umbral como función de activación y considerando la entrada ficticia en todos los casos:

A diagram of a triangle with letters and numbers

Description automatically generated

Determinar los valores de los pesos de la red neuronal para que clasifique como 1 la siguiente región geométrica:

(a)

(b)

En ambos casos, explica con detalle el razonamiento que te ha llevado a la conclusión.

(a)

Se hará que la neurona C simule el funcionamiento de una puerta lógica OR. Así, se puede trabajar por separado la salida de la neurona A y la B.

En la neurona A:

Asignemos que la neurona A debe cumplir con la condición:

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

Así:

En la neurona B:

Asignemos que la neurona B debe cumplir con la condición:

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

Así:

Finalmente queremos que la neurona C simula una compuerta lógica OR.

De la tabla de verdad de la compuerta OR:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Nos quedan las siguientes desigualdades:

Los valores:

Cumplen con todas las desigualdades.

Por lo que los valores finales de la red son:

(b)

Se hará que la neurona C simule el funcionamiento de una puerta lógica AND. Así, se puede trabajar por separado la salida de la neurona A y la B.

En la neurona A:

Asignemos que la neurona A debe cumplir con la condición:

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

Así:

En la neurona B:

Asignemos que la neurona B debe cumplir con la condición:

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

Así:

Finalmente queremos que la neurona C simule una compuerta lógica OR.

De la tabla de verdad de la compuerta AND:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Nos quedan las siguientes desigualdades:

Los valores:

Cumplen con todas las desigualdades.

Por lo que los valores finales de la red son: